

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ**

**ХАРКІВСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ  
МІСЬКОГО ГОСПОДАРСТВА**

## **МЕТОДИЧНІ ВКАЗІВКИ**

**ДО ПРОВЕДЕННЯ ПРАКТИЧНИХ ЗАНЯТЬ  
ТА САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ**

**З ДИСЦИПЛІНИ**

# **Інформаційні технології на транспорті**

(для студентів усіх форм навчання спеціальностей:  
7.05070202, 8.05070202 «Електричні системи і комплекси  
транспортних засобів»; 7.05070203, 8.05070203 «Електричний транспорт»  
та слухачів другої вищої освіти спеціальності «Електричний транспорт»)

**Харків  
ХНАМГ  
2011**

Методичні вказівки до проведення практичних занять та самостійної роботи студентів з дисципліни «Інформаційні технології на транспорті» (для студентів усіх форм навчання спеціальностей: 7.05070202, 8.05070202 «Електричні системи і комплекси транспортних засобів»; 7.05070203, 8.05070203 «Електричний транспорт» та слухачів другої вищої освіти спеціальності «Електричний транспорт») / Харк. нац. акад. міськ. госп-ва; уклад.: К. О. Сорока, Н. В. Гарбуз, О. С. Гордієнко. – Х.: ХНАМГ, 2011. – 56 с.

Укладачі: К. О. Сорока,  
Н. В. Гарбуз,  
О. С. Гордієнко

Рецензент: проф., д.т.н. В. Ф. Рой

Методичні вказівки побудовані за вимогами кредитно-модульної системи організації навчального процесу.

Рекомендовано кафедрою електричного транспорту,  
протокол № 5 від 23. 11. 2010 р.

## ЗМІСТ

Стор.

ПРАКТИЧНЕ ЗАНЯТТЯ № 1 .....	5
Ознайомлення з АРМ інженера тролейбусного депо .....	5
1.1. Опис пакета програм АРМ інженера тролейбусного депо .....	5
1.2. Введення АРМ інженера в експлуатацію.....	6
1.3. Робота з програмним пакетом .....	6
1.4. Порядок використання програмного пакета.....	8
1.5. Контрольні питання.....	9
ПРАКТИЧНЕ ЗАНЯТТЯ № 2. ....	10
Розробка технічного завдання для створення ІС чи АРМ .....	10
2.1. Послідовність розробки ІС.....	12
2.2. Визначення цілей та уточнення завдань ІС .....	12
2.3. Визначення послідовності виконання завдань .....	13
Контрольні питання .....	14
ПРАКТИЧНЕ ЗАНЯТТЯ №3. ....	15
Аналіз завдань та даних ІС. Розробка структури даних для реалізації інформаційної системи .....	15
Контрольні питання .....	18
ПРАКТИЧНЕ ЗАНЯТТЯ №4. ....	19
Розробка та оформлення логічної моделі даних .....	19
інформаційної системи .....	19
Контрольні питання .....	27
ПРАКТИЧНЕ ЗАНЯТТЯ №5. ....	29
Ознайомлення з CASE-технологіями. Побудова контекстної діаграми та функціональної моделі IDEF1 .....	29
5.1. Опис функціональної моделі .....	30
5.2. Вимоги до порядку побудови моделі.....	33
5.3. Побудова функціональної моделі системи (моделі IDEF0) .....	35

ПРАКТИЧНЕ ЗАНЯТТЯ №6. ....	40
Розробка запитів та звітів для аналізу даних ІС (АРМ) .....	40
Контрольні питання .....	41
ПРАКТИЧНЕ ЗАНЯТТЯ №7. ....	42
Розробка форм, автоматизація створеної моделі ІС (АРМ).....	42
ПРАКТИЧНЕ ЗАНЯТТЯ №8. ....	44
Створення презентацій для демонстрації роботи .....	44
створеної ІС (АРМ) .....	44
8.1. Створення першої сторінки презентації.....	45
8.2. Створення наступних слайдів презентації .....	46
8.3. Створення сторінки без авторозмітки .....	46
8.4. Підготовка презентації до показу. Анімація слайда .....	49
Контрольні питання .....	50
ПРАКТИЧНЕ ЗАНЯТТЯ № 9. ....	51
Реалізація АРМ в комп'ютерній мережі .....	51
Архітектура комп'ютерних мереж.....	51
9.2. Стандарти рівнів відкритих телекомунікаційних систем.....	52
Контрольні питання .....	54

# **ПРАКТИЧНЕ ЗАНЯТТЯ № 1**

## **Ознайомлення з АРМ інженера тролейбусного депо**

### **1.1. Опис пакета програм АРМ інженера тролейбусного депо**

Пакет програм АРМ інженера тролейбусного депо призначений для ведення обліку пробігу рухомого складу і виконання планових і позапланових ремонтів. Витрати часу на ведення даної інформації складають не більше 30 хв. на день для парку на 200 одиниць рухомого складу.

Пакет програм дозволяє одержувати різні довідки щодо ремонтів і пробігів рухомого складу.

Програми були згруповані за чотирма технологічними ознаками:

- ведення довідкових баз даних;
- щоденне введення даних;
- отримання довідок і звітів;
- корегування пакета і обслуговування баз даних.

Пакет прикладних програм інженера тролейбусного депо з обліку роботи рухомого складу складається з трьох частин:

1. Облік пробігів і ремонтів рухомого складу;
2. Облік відмов, несправностей, вибувань рухомого складу;
3. Облік роботи покришок на рухомому складі (може використовуватися і для ведення обліку роботи й інших вузлів рухомого складу, що мають індивідуальні номери, система обліку яких аналогічна системі обліку роботи покришок).

Використання АРМА інженера дає можливість,:

- складати графік планових ремонтів;
- прогнозувати ремонти на черговий рік;
- одержувати різні довідки про пробіги рухомого складу;
- одержувати довідки про простій в планових і непланових ремонтах;
- аналізувати причини постановок в непланові ремонти.

## **1.2. Введення АРМ інженера в експлуатацію**

Для запуску пакета необхідно скопіювати файл TOD4.arj і два файли \*.hlp в каталог, де буде знаходитись пакет програм і створити підкаталоги з іменами, як показано на дискеті. Потім в каталог HELP скопіювати файли «Допомоги».

Після розархівзації пакета при першому запуску необхідно вибрати опцію СЕРВІС, а в ній програму ПОЧАТКОВІ УСТАНОВКИ і там задати шляхи пошуку інформації.

Наступні кроки вводу в експлуатацію пакета такі:

1. Вибрати опцію СЕРВІС, натиснути клавішу Enter, вибрати програму НАЛАШТУВАННЯ ПАКЕТА, ввести необхідну інформацію. За необхідності натиснути клавішу F1 і ознайомитися з інструкцією з налаштування пакета.

2. Вибрати опцію ДОВІДКОВІ БАЗИ ДАНИХ і ввести довідкову інформацію. Введення інформації потрібно проводити в порядку проходження програм в меню опції ДОВІДКОВІ БАЗИ ДАНИХ. Інструкції з введення інформації викликаються при натисненні клавіші F1 в кожній з програм.

3. Вибрати опцію СЕРВІС, вибрати з меню опції програму ПРОБІГИ ВІД РЕМОНТІВ і ввести дані про дати випуску і пробіги рухомих одиниць.

4. Вибрати опцію СЕРВІС, вибрати з меню опції програму НОМЕРИ РЕМОНТІВ і ввести для кожної РО кількість ремонтів виконаних щодо ремонту вищої категорії.

5. Після закінчення введення вказаної вище інформації вивести її на друк і виконати звірку даних. виправити виявлені помилки введення. Після виконання робіт по пунктам 1-5 пакет програм готовий до експлуатації.

## **1.3. Робота з програмним пакетом**

Основою роботи АРМ є супровід бази даних шляхом щоденного введення даних щодо експлуатації тролейбусів у депо.

Для цього призначені команди меню з розділу ЩОДЕННЕ ВВЕДЕННЯ ДАНИХ. Вони викликають програми, які використовуються щодня для

введення інформації в основні бази даних. Щоденне введення інформації проводиться в порядку проходження програм в меню опції.

Решта опцій головного меню пакета використовується за ступенем необхідності. Кожна програма викликається з меню опцій, має інструкцію з роботи. Інструкції по роботі з програмами викликаються після натискання клавіші F1.

### **1.1.1. Склад програмного пакету**

Основу програмного пакету АРМ інженера складають бази даних з 25 таблицями. Основні з них такі:

Список водіїв	Коди ремонтних робіт
Список ремонтного персоналу	Кодифікатор деталей
Рухомий склад	Кодифікатор вузлів
Облік рухомих одиниць	Кодифікатор обладнання
Типи рухомого складу	Графік ремонтів
Маршрути	Планові ремонти
Облік пробігів рухомого складу	Вибуття рухомого складу
Види ремонтів і їх основні показники	Відмови рухомого складу
Міжремонтні пробіги	Простої в депо
Виконані планові ремонти	Облік відмов рухомого складу
Причини позапланових ремонтів	Пробігу рухомого складу
Кодифікатор видів вибувань	Прогноз ремонтів на рік
Кодифікатор причин відмов	Введення непланових ремонтів
Місця усунення відмов	Введення позапланових ремонтів
Місця розташування вузлів	Введення добового пробігу
	Виконані планові ремонти

На основі даних, занесених до таблиць, виконуються такі розрахунки та аналізи:

Аналіз відмов.

Причини позапланових ремонтів.

Довідка про перебування в позапланових ремонтах.

Довідка про перебування в планових ремонтах.  
Пробіг від ремонтів.  
Аналіз замін вузлів і деталей.  
Коефіцієнт трудової участі.  
Порівняння двох періодів роботи.  
Розподіл вибуттів за місяцями.  
Коефіцієнт використання рухомого складу.  
Кількість позапланових ремонтів за номерами рухомих одиниць.  
Підсумок причин позапланових ремонтів.  
Надійність рухомого складу.  
Надійність вузлів і деталей.

Для виводу результатів аналізу розроблена система різних типів бланків, звітні форми довідки.

#### **1.4. Порядок використання програмного пакета**

Для введення програмного пакета в дію після його встановлення на комп'ютери потрібно ввести інформацію в базу даних щодо персоналу депо, а саме водіїв з їх персональними даними, ремонтного персоналу і всіх інших працівників, задіяних в роботі депо, а також щодо наявних в депо рухомих одиниць, їх пробігів та стану на момент встановлення програмного забезпечення. Це досить кропітка частина роботи, яка вимагає багато часу. Для її полегшення можна використати наявні електронні таблиці, які є в депо на момент введення в дію АРМ.

Експлуатація програмного пакета полягає у щоденному введенні в базу даних комп'ютера результатів роботи депо за поточний робочий день, витрати часу на виконання цієї роботи складають приблизно 30 хв. робочого часу оператора. Оператор вводить дані про щоденні пробіги рухомого складу, виконане обслуговування транспортних засобів, виконані планові ремонти,



заявки на позапланові ремонти, результати виконаних ремонтів, заміни деталей та вузлів в окремих тролейбусах та інші дані.

Результати роботи депо виводяться за ступенем необхідності. В першу чергу, щодня виводяться результати роботи за минулий та поточний день, виводяться дані помісячної роботи і усі потрібні для керування роботою депо довідки та результати аналізу.

Під час виконання практичних занять для ознайомлення з роботою АРМ студенти повинні розглянути таблиці, бланки на звіти, які підтримує програмний пакет. Студенти також вводять дані про результати роботи депо в окремі таблиці пакета. Для інформації про порядок введення даних та формати, в яких дані потрібно вказувати, потрібно скористатись клавішею F1, за допомогою якої викликається інструкція по кожному з потрібних пунктів.

### **Контрольні питання**

1. У базі депо наведено характеристики 400 рухомих одиниць, восьми типів. Опис даних одного типу РО займає 20 Кбіт, а індивідуальних характеристик однієї РО – 10 Кбіт. Який обсяг бази, приведеної до ЗНФ.
2. Підрахуйте обсяг бази даних 1200 співробітників в Кбайтах з їх персональними даними, фотокарткою, автобіографією. Поля бази даних виберіть самостійно. Для фотокартки потрібно 70 Кбайт інформації.

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 2**

### **Розробка технічного завдання для створення ІС чи АРМ**

Для створення ІС, у першу чергу, потрібно визначити завдання, які вирішуватиме система, і з якими об'єктами вона працюватиме. На основі цього створюється база даних (БД), у яку увійшли б усі потрібні дані. Створення бази даних розпочинається з їх опису.

У будь-якій інформаційній системі розрізняють постійні та змінні дані. Постійні дані потрібні для підтримки функціонування ІС. Вони складають основу ІС і є довідниковими для виконання певних функцій, покладених на ІС. Як правило, ці дані є найбільш загальні та використовуються для вирішення цілої низки завдань. В якості постійних даних використовують дані про кадри та склад системи, про матеріально-технічну базу та основне обладнання.

Опис даних здійснюється на основі аналізу предметного складу системи, який виконано у першій частині роботи. Для кожного об'єкта, який визначено, що він входить у склад системи і відіграє суттєву роль в створеній ІС, потрібно визначити основні характеристики та виписати їх для наступного вводу у базу даних. Дані про кадровий склад можна отримати зі штатного розкладу підприємства чи його підрозділів. Які саме дані потрібно відібрати визначається призначенням ІС. Як правило, потрібно використовувати якомога повніші дані. Наприклад, при початковій розробці ІС може видатись, що дані про вік співпрацівників, час, коли вони почали роботу на підприємстві, не потрібен. Пізніше, коли розпочинається розробка ІС, виявляється, наприклад, що для розрахунку заробітної плати потрібно враховувати вік працівника та стаж роботи, від яких залежать розміри премії та пільги працівника. Тому під час початкового збору інформації потрібно відбирати якомога повнішу інформацію. Інформацію слід відбирати з документів, для того, щоб гарантувати її достовірність.

Відомості про обладнання, яке задіяне у технологічному процесі, слід також збирати якомога повніше. Під час збору інформації потрібно, як правило, використовувати паспортні дані обладнання або результати його паспортизації, атестації, періодичної перевірки. Зібрані дані слід оформлювати у вигляді сукупності таблиць.

У випадку створення ІС для підтримки певних технологічних процесів обслуговування інженерних мереж, транспорту в якості постійних даних вносять відомості про мережі, їх протяжність, інженерне обладнання, рухомий склад, його типи, дати ремонтів, пробіги на певний період часу. Дані окремих типів обладнання вносять в окремі таблиці, згідно паспортів. При формуванні бази даних транспортної мережі вказуються дані протяжності ліній, зупиночних пунктів, відстаней між зупинками, відстані від депо до певної ділянки лінії тощо. При відборі даних про маршрутну систему вносять послідовності зупиночних пунктів, а також дані про резервні маршрути, які можуть бути використані в разі виникнення позаштатних ситуацій. Аналогічні дані відбираються і щодо інженерних мереж. Зібрані дані оформлюють у вигляді сукупності таблиць.

Змінні дані, як, наприклад, результати експлуатації, пробіги РО, щомісячна чи щоденна оплата і т.п., не обов'язково повинні бути записані у повному обсязі. Потрібно тільки визначити, яка структура змінних даних, їх зміст, обсяг. Треба визначити методи, за допомогою яких збирають ці дані та вводять в бази даних. У всіх випадках бажано автоматизувати процес введення змінних даних. Для цього можуть використовуватись датчики, вимірювальні прилади, які фіксують результати роботи на електронних носіях інформації. Можуть використовуватись пристрої, що полегшують введення даних, наприклад, автоматичні зчитувачі інформації з маршрутних листків рухомих одиниць тощо.

Постійні та змінні дані завжди потрібно прив'язувати до конкретних об'єктів, які задіяні в технологічному процесі.

Звітним матеріалом з виконаної роботи по даному етапу є сукупність таблиць з введеною у них інформацією.

## **2.1. Послідовність розробки ІС**

1. Визначення цілей та уточнення завдань ІС (АРМ).
2. Визначення послідовності виконання завдань.
3. Аналіз даних, потрібних для роботи ІС.
4. Визначення структури та побудова логічної моделі даних.
5. Визначення типів звітної документації, запитів, форм, алгоритмів обробки інформації.
6. Розробка бази даних на комп'ютері.
7. Створення запитів, форм, звітів.
8. Оформлення прикладного пакета ІС.
9. Тестування та вдосконалення ІС.

Перші п'ять етапів виконуються без застосування комп'ютера, проте їх роль при розробці ІС є визначальною. На ці етапи витрачається, як правило, 50% часу, витраченого на всю розробку ІС. Ефективність використання ІС після її розробки визначається на цих початкових етапах розробки. Від того, наскільки якісно вони будуть виконані, залежить і швидкість створення ІС (АРМ), і зручність її використання. Розглянемо детальніше ці етапи і зміст роботи під час їх виконання.

## **2.2. Визначення цілей та уточнення завдань ІС**

Мета, для якої призначена ІС (АРМ), що потрібно створити, оговорено в темі роботи. Для досягнення цієї мети потрібно вирішити кілька конкретних завдань, які є допоміжними або доповнюють основну ціль. Ці завдання потрібно сформулювати самостійно.

Для прикладу розглянемо процедуру уточнення завдань для ІС, яка призначена для ведення обліку пробігу тролейбусів. Такий облік потрібен для контролю роботи рухомого складу, для складання графіків ремонтів, для списання РО після завершення експлуатації. Ці дані також є вихідними даними, що відображають результат роботи депо. Залежно від конкретного депо,

процедура випуску тролейбусів на маршрути складається: з підготовки транспортної одиниці до випуску; одержання водієм маршрутного листка із завданням на робочий день; виїзду з депо і прямування до лінії маршруту; виконання визначеної кількості рейсів вздовж маршруту; повернення в депо і здача тролейбуса та маршрутного листка. Маршрутний листок під час роботи на маршруті заповнюють, як правило, диспетчери кінцевих станцій. Виходячи з цього, ІС вона повинна виконувати, наприклад, такі завдання:

- 1) реєстрацію випуску РО з депо;
- 2) реєстрацію повернення РО в депо;
- 3) реєстрацію водіїв та випуск їх на маршрути;
- 4) зчитування інформації, занесеної в маршрутні листки;
- 5) обрахунок пробігів РО;
- 6) визначення сумарних пробігів за певні інтервали часу;
- 7) аналіз пробігів та підготовку даних для складання графіка планових ремонтів:
- 8) видачу протоколів пробігу РО на вимогу керівництва;
- 9) підготовку звітів роботи рухомих одиниць за місяць; за квартал; за рік;
- 10) видачу підсумків за результатами роботи депо.

Додатковими завданнями можуть бути: завдання обліку та проходження планових ремонтів, обліку несправностей РО, обліку роботи водіїв тощо.

### **2.3. Визначення послідовності виконання завдань**

Послідовність виконання завдань можна відобразити у вигляді списку або за допомогою діаграми, як показано на рис. 1.

Як видно з діаграми, певну частину завдань складають розрахунки потрібних величин. Як правило, в інформаційних системах розрахунки включають у процедури підготовки протоколів чи звітів. Тому завдання розрахунків слід включати до завдань видачі протоколів чи звітів. На схемі (рис. 1) це показано пунктирними лініями.



Рис. 1 – Приклад діаграми послідовності виконання завдань ІС

### Контрольні питання

1. В чому переваги та недоліки каскадної моделі розробки інформаційної системи. Зобразіть її.
2. В чому переваги та недоліки спіральної моделі розробки інформаційної системи. Зобразіть її.
3. Зобразіть приклад логічної моделі диспетчера депо з випуску тролейбусів на маршрути.

## ПРАКТИЧНЕ ЗАНЯТТЯ №3

### Аналіз завдань та даних ІС. Розробка структури даних для реалізації інформаційної системи

Після визначення завдань ІС виконують аналіз даних, які потрібні для виконання кожного завдання. Для цього використовують результати субстрактного аналізу системи. Дані потрібно занести в зразки таблиць 1 та 2.

Таблиця 1 – Зразок бланка даних ІС

Бланк опису завдань № 1			
<b>Назва завдання:</b>	Реєстрація випуску і прибуття РО		
<b>Зміст:</b>	Виконується диспетчерами з випуску. Дані заносять в журнал диспетчера (комп'ютер)		
<b>Список логічно пов'язаних завдань:</b>	Реєстрація прибуття в депо Реєстрація роботи водіїв Видача та прийом маршрутних листків		
Елемент даних (Атрибут)	Об'єкт (Сутність)	Використання	Опис
Дата	Журнал диспетчера	I	Ввід з клавіатури
Номер РО	Тролейбуси	I	Вибір зі списку
Водій	Співробітники	I	Вибір зі списку
Маршрут	Маршрути	I	Вибір зі списку
Час випуску	Журнал диспетчера	I	Ввід з клавіатури
Час прибуття	Журнал диспетчера	I	Ввід з клавіатури
№ марш. листка	Маршрутний листок	I	Ввід з клавіатури
Диспетчер	Співробітники	I	Вибір зі списку

В стовпчик таблиці «Елемент даних» вносять опис того атрибута даних, який використовується для виконання завдання. В стовпчик таблиці «Об'єкт» вноситься назва об'єкта якому належить цей атрибут. В стовпчик «Використання» записують скорочене позначення, для чого в даному завданні використовується цей атрибут даних, а саме: I – ввід даних (Input), O – вивід даних (Output), U – заміна даних (Utput), D – вилучення даних (Delete), C – розрахунок (Calculate). В стовпчик «Опис» вносять опис способу використання атрибута для виконання завдання.

Такі бланки складають для виконання всіх завдань ІС.

Таблиця 2 – Зразок бланку завдань

Бланк опису завдань № 7			
<b>Назва завдання:</b>	Видача протоколів щоденних пробігів		
<b>Зміст:</b>	Підрахунок пробігів тролейбусів в км та видача протоколу		
<b>Список логічно пов'язаних завдань:</b>	Видача протоколів місячних пробігів		
Елемент даних (Атрибут)	Об'єкт (Сутність)	Використання	Опис
Дата	Журнал диспетчера	I	Ввід з клавіатури
Номер РО	Тролейбуси	I	Вибір зі списку
№ маршруту	Маршрути	I	Вибір зі списку
Довжина оборотного рейсу	Маршрути	I	Вибір зі списку ( $L_{об}$ )
Кількість оборотних рейсів	Марш. листок	I	Ввід з клавіатури ( $N$ )
Довжина нульового пробігу	Маршрути	I	Вибір зі списку ( $L_0$ )
Підрахунок пробігу	Запит	C	$P=N \cdot L_{об} + 2 \cdot L_0$ *
Сумарний пробіг	Протокол	O	Роздрук

\* Тут:  $P$  – пробіг РО,  $N$  – кількість рейсів,  $L_{об}$  – довжина оборотного рейсу,  $L_0$  – довжина нульового пробігу.

Після виконання опису завдань та складання таблиць завдань потрібно скласти список об'єктів (сутностей), яких стосуються дані. Під сутностями розуміють категорії, які мають суттєве значення в даній ІС. У нашому випадку це можуть бути люди, які виконують певні обов'язки, наприклад, сутності «Співробітники», «Водії», «Диспетчери», «Бригади», «Цехи» і т.п.; певні організації: «Депо», «Насосна станція», «Метрополітен», «Підприємство» і т.п.; певне обладнання: «Насоси», «Тролейбуси» тощо; певні документи: «Замовлення», «Маршрутний листок», «Журнал реєстрації», «Інструкція», «Накази» усні чи письмові. Тобто сутності – це узагальнене поняття об'єкта, оскільки сутностями є, наприклад, «Бригади», «Відомості закріплення тролейбусів», «Списки телефонів», «Накази», що важко назвати об'єктом, але



вони відіграють важливу роль в конкретній системі, й саме до них слід співвідносити дані.

Після складання списку сутностей для кожної з них будують таблицю на зразок табл. 3.

Таблиця 3 – Зразок бланку даних

<b>Бланк опису даних № 1</b>			
<b>Назва сутності (об'єкта) :</b>	Журнал диспетчера		
<b>Короткий опис:</b>	Записи в журналі випуску (в електронній формі)		
<b>Пов'язані сутності:</b>	<b>Характеристики зв'язку</b>		
	<b>Назва</b>	<b>Ступінь зв'язку</b>	
	Тролейбуси	Багато до одного	
	Водії	Багато до одного	
	Маршрутні листки	Багато до одного	
<b>Елемент даних (Атрибут)</b>	<b>Тип даних</b>	<b>Опис</b>	<b>Умова на значення</b>
Дата випуску	Дата/час	Поточна дата	Звичайна для дати (вводиться автоматично)
Час випуску	Дата/час	Час виїзду з депо	Звичайна для часу
Час повернення	Дата/час	Час повернення в депо	Звичайна для часу
№ маршруту	Маршрути	На який маршрут	Існуючі маршрути
Водій	Текстовий (30)	Прізвище та ініціали	Прізвища водіїв
Диспетчер випуску	Текстовий (30)	Прізвище та ініціали	Прізвища водіїв (значення за замовчанням)

В стовпчику «Пов'язані сутності» записують назви сутностей, з якими повинна бути пов'язана дана сутність. В стовпчику «Ступінь зв'язку» записують ступінь зв'язку на обох сторонах зв'язку, а саме, по-перше, – на стороні даної сутності та, по-друге, – на стороні пов'язаної з нею сутності. Наприклад, в журналі диспетчера може бути багато записів про випуск певного тролейбуса, але кожний запис в журналі відноситься тільки до одного тролейбуса.

В стовпчику таблиці «Елемент даних (Атрибут)» вносять назву елемента даних (атрибута), який вводять в базу даних. В стовпчику таблиці «Тип даних» вносять тип даних та максимальне значення розміру поля, виділеного для елемента даних. В стовпчику «Опис» записують короткий опис елементів даних, які будуть вводиться в БД. В стовпчику «Умова на значення» вносяться додаткові умови або обмеження, яким повинне відповідати значення з метою запобігання можливих помилок.

Такі бланки обов'язково складають для виконання всіх сутностей ІС.

### **Контрольні питання**

1. Зобразіть сутність, не приведену до ЗНФ. Вкажіть їх атрибути.
2. Зобразіть сутність зі складним ключем. Вкажіть їх атрибути.
3. Зобразіть дві сутності, між якими встановлено зв'язок один до одного.  
Вкажіть їх атрибути.
4. Зарисуйте дві сутності зі зв'язком один до багатьох. Вкажіть їх атрибути.

## ПРАКТИЧНЕ ЗАНЯТТЯ №4

### Розробка та оформлення логічної моделі даних інформаційної системи

У заповнених бланках завдань фактично розміщена вся інформація, потрібна для побудови ІС. Після збору даних можна приступати до безпосереднього їх вводу в комп'ютер та створювати ІС. Проте такі дані, як правило, не дозволяють створити ефективної ІС. Вони можуть бути недостатніми чи нерационально розміщеними. Для впорядкування даних, встановлення зв'язків між ними будується логічна модель даних.

Логічна модель даних – це загальний погляд на дані в певній предметній області.

У логічній моделі, відповідно до типу бази даних, а саме реляційної БД, дані розміщують у сутностях, які об'єднані між собою певними зв'язками. Ці зв'язки відповідають логіці функціонування реальної системи. Логічну модель даних інколи називають концептуальною. Створюється логічна модель даних відповідно до правил, які стандартизовані на міжнародному рівні. У цій роботі створюється модель, яка має назву «Сутність – зв'язок». Її загальноприйняте позначення ERD – модель або ER – Diagramming (E – Essences – сутності, R – Relation – відношення, зв'язки). Ця модель відповідає міжнародному стандарту IDEF 1x. Побудова логічної моделі даних (ERD – моделі) виконується у такому порядку:

- виділяють сутності та їх атрибути;
- визначають ключові атрибути;
- зображають сутності на аркуші паперу чи у електронній формі (в разі використанні CASE-технологій);
- встановлюють зв'язки між сутностями та визначають характеристики зв'язку.
- виконують нормалізацію логічної моделі даних.

1. Побудова логічної моделі розпочинається з виділення сутностей. Поняття сутностей визначено дещо вище. Для кожної сутності складена таблиця атрибутів.

*Зауваження: Під час виділення сутностей слід враховувати, що однотипні дані не повинні повторюватись у таблиці сутності. Наприклад, виділивши окремо сутність «Тип тролейбуса», ми усуваємо необхідність для кожного тролейбуса у сутності «Тролейбуси» вказувати паспортні дані.*

2. Наступним кроком після опису завдань та заповнення бланків завдань є опис об'єктів бази даних. Усі дані ІС повинні бути прив'язані до певних об'єктів (сутностей). Під сутностями розуміють певні об'єкти, категорії, які мають суттєве значення у даній предметній області, системі. Це можуть бути люди, які виконують певні обов'язки, наприклад, «Співробітники», «Водії», «Диспетчери», «Бригади», «Цехи» і т.п., певні організації: «Депо», «Насосні станції», «Метрополітен», «Підприємства» і т.п., певне обладнання: «Насоси», «Тролейбуси» тощо; певні документи: «Замовлення», «Маршрутний листок», «Журнал реєстрації», «Інструкція», «Накази», «Розпорядження» письмові чи усні. Кожній сутності у базі даних відповідає таблиця.
3. Для кожної сутності визначають ключові атрибути. Ключовим є атрибут, який однозначно визначає кожен екземпляр сутності. Ключовим атрибутом сутності «Тролейбуси» може бути «номер» тролейбуса. Кожному тролейбусу в депо присвоюється свій обліковий номер, і він є індивідуальним. Для сутності «Водії» ключовим атрибутом може бути «прізвище». Проте в депо можуть бути тезки за прізвищем, а такий атрибут не дозволить однозначно визначити водія. Більш визначальним є складний атрибут, в який входять атрибути: «прізвище», «ім'я» та «по батькові». Складним атрибутом користуватись незручно і, як правило, вводять додатковий атрибут, як, наприклад, «табельний номер».

4. На аркуші паперу зображають усі сутності, які повинні входити у базу даних. Сутності зображають прямокутником, над яким записують назву сутності. Всередині сутності записують її атрибути. Ключові атрибути відділяють лінією від інших атрибутів. Зразок зображення сутностей наведено на рис. 2.

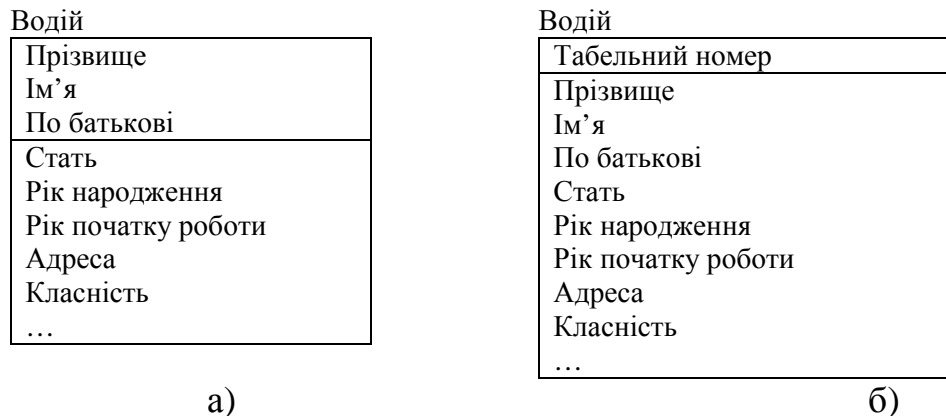


Рис. 2 – Приклади зображення сутності «Водій»:

*а) зі складним ключем; б) з простим ключем*

5. Встановлюють зв'язки між сутностями. Зв'язки проводять олівцем. Зв'язки між сутностями повинні відповідати логічним зв'язкам, які існують у предметній області для якої створюється ІС. Зв'язки відповідають дієсловам. Наприклад, водій **керує** тролейбусом. Зв'язки проводять від початкової сутності до підпорядкованої. Усі сутності повинні бути певним чином пов'язані між собою.

Проставляють характеристики зв'язку. Зв'язок відповідає певному відношенню між сутностями. Зв'язок має такі характеристики, як ступінь (потужність) зв'язку, клас належності та тип зв'язку.

- а. Ступінь зв'язку може бути один до одного 1:1, один до багатьох 1:∞, багато до одного ∞:1 та багато до багатьох ∞:∞. Ступінь зв'язку позначається на кожному кінці лінії зв'язку відповідним символом. Якщо, наприклад, в депо встановлена однозначна відповідність тролейбусів і водіїв,

тобто кожен тролейбус закріплений за одним єдиним водієм, то ступінь зв'язку 1:1, якщо за водієм закріплений тільки один тролейбус, а за кожним тролейбусом закріплено 2 чи більше водіїв, то ступінь зв'язку на стороні водія  $\infty$ , а на стороні тролейбуса 1. Коли навпаки – за одним водієм закріплено декілька тролейбусів, але кожен тролейбус закріплено тільки за одним водієм, то ступінь зв'язку 1: $\infty$ . Якщо в депо закріплюється за одним водієм декілька тролейбусів, і за кожним тролейбусом закріплюється декілька водіїв, то ступінь зв'язку між сутностями «водії» та «Тролейбуси» –  $\infty:\infty$ . У більшості баз даних не допускається ступінь зв'язку  $\infty:\infty$ . Для усунення такого зв'язку вводять проміжну сутність, з якою встановлюється зв'язок 1: $\infty$  з обох сторін. Це може бути сутність «Відомість закріплення тролейбусів». У цьому разі зв'язок типу  $\infty:\infty$  переводиться в два зв'язки типу 1: $\infty$  і  $\infty:1$ . Конкретний водій (екземпляр сутності «Водій») може мати декілька записів у відомості закріплення, зв'язок – 1 :  $\infty$ . Кожен тролейбус також може мати декілька записів у відомості закріплення, тобто тут зв'язок типу 1: $\infty$ , а якщо його читати зі сторони відомості, то це відповідає зв'язку типу  $\infty:1$ .

б. Клас належності визначає, чи обов'язковим є зв'язок. Обов'язковий в класі належності означає, що кожен екземпляр сутності повинен обов'язково брати участь у зв'язку. За кожним водієм депо обов'язково закріплено один чи декілька тролейбусів. Необов'язковий клас належності означає, що не кожен об'єкт сутності бере участь в зв'язку. Наприклад, у депо є змінні водії, які не мають закріплених за ними тролейбусів. Необов'язковий клас належності позначається знаком «ромб» біля відповідної сутності. У розглянутому прикладі значок «ромб» потрібно вказати біля сутності «Водій».

с. Тип зв'язку може бути ідентифікаційний та неідентифікаційний. Ідентифікаційний тип зв'язку визначає, що батьківська сутність повністю визначається за дочірньою сутністю. Якщо розглядати сутності «Водій» та «Заявка на ремонт», то зв'язок є ідентифікаційним. Заявка на ремонт може бути подана тільки певним водієм. У випадку ідентифікаційного зв'язку дочірня сутність зображається прямокутником із заокругленими кутами. У випадку

ідентифікаційного зв'язку у базі даних ключ батьківської сутності переходить (мігрує) до складу ключа дочірньої сутності, це означає, що в сутності «Заявка на ремонт» потрібно ввести атрибут «Хто подає», і цей атрибут є також ключовим. Він має назву зовнішнього ключа і позначається символом FK. Ключовий атрибут бажано позначати символом К.

Зв'язок сутностей «Тролейбус» та «Водій» є неідентифікаційними, оскільки тролейбус не може однозначно ідентифікувати водія. При встановленні неідентифікаційного зв'язку ключ батьківської сутності також мігрує до дочірньої, але він додається як атрибут в список атрибутів.

*Зауваження. На першому етапі створення моделі даних допускаються деякі неточності в установленні характеру зв'язків, оскільки сама база даних, в деяких випадках, може самотійно проставляти певні типи зв'язків та виконувати процедуру міграції полів. Проте бажано з самого початку створення бази даних уточнити характер зв'язків. Особливо це легко зробити для встановлення ідентифікаційного зв'язку між сутностями та ступеня (потужності) зв'язку.*

6. Виконують нормалізацію логічної моделі даних. Нормалізація полягає в послідовному приведенні структури даних (логічної моделі) до якомога вищої нормальної форми. Дані послідовно приводяться до першої 1НФ, другої 2НФ, третьої 3НФ та вищих нормальних форм. Відомо шість нормальних форм організації даних, а саме 1НФ, 2НФ, 3НФ, підсилена 3НФ, 4НФ і 5НФ. Як правило, достатньо привести логічну модель до 3НФ.

**Перша нормальна форма.** Сутність знаходиться в першій нормальній формі (1НФ) тоді, коли всі атрибути сутності є атомарними (простими) і немає груп атрибутів, що повторюються. Для приведення сутності до 1НФ треба:

- 1) розділити складні атрибути на прості;
- 2) виділити усі атрибути, які повторюються;
- 3) створити нову сутність;
- 4) перенести в нову сутність атрибути, що повторювались;

5) встановити ідентифікаційний зв'язок від початкової сутності до створених.

Наприклад, привести сутність, показану на рис. 3, до першої нормальної форми.

Співробітник

Табельний номер
Прізвище, ім'я, по батькові
Кафедра, посада
Назва предмета, який веде
Оклад
Телефон
Дата зарахування і звільнення

Рис. 3 – Приведення сутностей до першої нормальної форми (1НФ)

Тут, у сутності “Співробітник”, прізвище, ім'я і по батькові вважаються одним атрибутом, аналогічно з місцем роботи – кафедра та посада, дата зарахування та звільнення. Ці сутності є складними і їх треба розбити на прості. Крім цього, співробітник може вести декілька предметів і мати декілька номерів телефонів. На рис.4 показано цю сутність після першого кроку приведення до 1НФ.

Співробітник

Табельний номер
Прізвище
Ім'я
По батькові
Дата зарахування
Кафедра
Посада
Оклад
Предмет, який веде 1
Предмет, який веде 2
Предмет, який веде 3
Телефон 1
Телефон 2
Дата звільнення

Рис. 4 – Сутність після першого кроку приведення до 1НФ

Тут всі атрибути є простими (атомарними). Але є дві групи атрибутів, що майже співпадають: це предмети, які веде співробітник, і номери телефонів. Співробітник може вести не три, а інше число предметів, він же може мати не



два телефони (мобільний і звичайний), а ще телефон у родичів, де він часто буває. На наступних кроках приведення до 1НФ потрібно ввести нові сутності. Результат такого приведення показаний на рис. 5. Крім сутності “Співробітник”, створені ще дві сутності – “Предмет” і “Телефон”.

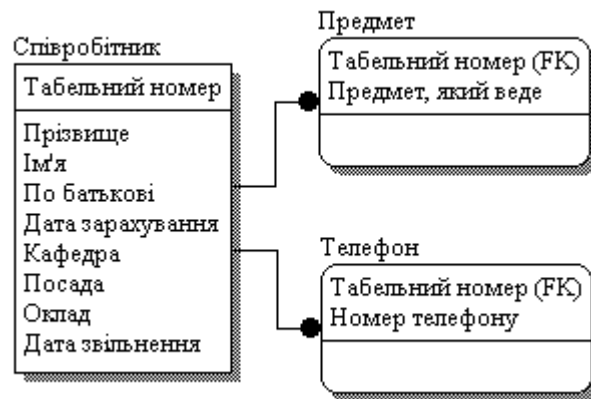


Рис. 5 – Сутність “Співробітник”, приведена до 1НФ

**Друга нормальна форма.** Сутність знаходиться у другій нормальній формі (2НФ) тоді, коли вона знаходиться в 1НФ і всі неключові атрибути функціонально повністю залежать від первинного ключа. Можливі випадки, коли в сутності є складний ключ і деякі атрибути залежать тільки від одного чи декількох, але не всіх одночасно атрибутів ключа. У цьому випадку для приведення до 2НФ потрібно:

- 1) виділити атрибути, що залежать тільки від частини первинного ключа;
- 2) створити нову сутність і помістити в них виділені атрибути;
- 3) встановити ідентифікаційний зв'язок від початкової сутності до створених.

Цей приклад показано на рис. 6.

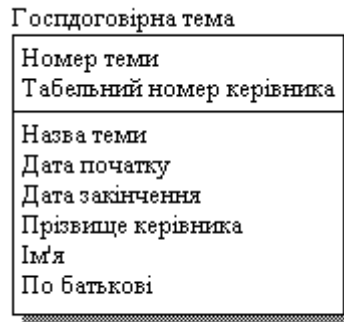


Рис. 6 – Приклад сутності “Госпдоговірна тема”

Тут прізвище керівника, його ім'я і по батькові залежать тільки від частини ключа, а саме від табельного номера, сутність не відповідає 2НФ. Результат приведення до 2НФ показано на рис. 7. Із сутності “Госпдоговірна тема” виділена сутність «Керівник теми».

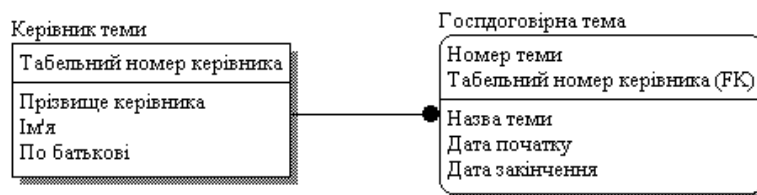


Рис. 7 – Сутність “Госпдоговірна тема”, приведена до 2НФ

**Третя нормальна форма.** Сутність знаходиться в 3НФ, якщо вона знаходиться в 2НФ і ніякий неключовий атрибут не залежить від іншого неключового атрибута. Наприклад, на рис. 5 у сутності “Співробітник” оклад залежить від посади і надбавки за вислугу років, тобто від двох неключових атрибутів, а саме посади й дати зарахування. Для приведення сутності до 3НФ потрібно:

- 1) створити нову сутність і перенести в неї атрибути, що мають одну і ту ж залежність від неключового атрибута;
- 2) використати атрибути, що визначають залежність, як ключі нової сутності;

3) встановити неідентифікаційний зв'язок від нової сутності до старої.

Приклад приведення до 3НФ показаний на рис. 8. Атрибут “Оклад” з даної сутності виведено. Оскільки оклад складається з тарифної ставки і доплати за стаж, то створена нова сутність, що визначає оклад працівника. Після цього сутність “Співробітник” приведена до 3НФ.

**Четверта нормальна форма 4НФ** вимагає, щоб була відсутня багатозначна залежність між атрибутами. Як правило, обмежуються третьою нормальною формою.

Після зображення усіх сутностей, встановлення зв'язків між ними та приведення сутностей до 3НФ логічну модель бази даних можна вважати завершеною.

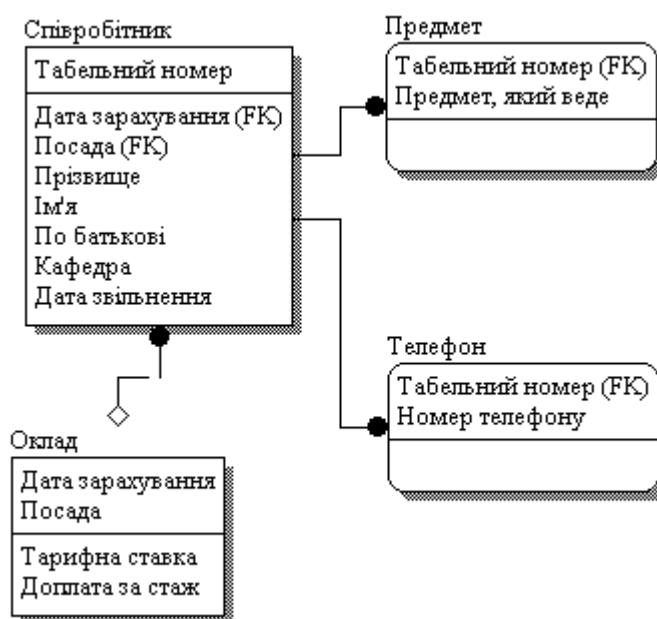


Рис. 8 – Приклад приведення сутності “Співробітник” до 3НФ

### Контрольні питання

1. Наведіть приклад і продемонструйте порядок приведення сутності до 1НФ.
2. Наведіть приклад і продемонструйте порядок приведення сутності до 2НФ.
4. Зобразіть сутність зі складним ключем, не приведену до 3НФ.
5. Наведіть приклад і продемонструйте порядок приведення сутності до 2НФ.
6. Наведіть і зобразіть таблицю сутності з аномалією вилучення даних.

7. Зобразіть сутність з п'ятьма атрибутами та відповідну їй таблицю.
8. На яку величину зміниться обсяг бази даних 1200 співробітників, якщо для запису автобіографії використати поле МЕМО замість символьного поля бази даних? Середній розмір автобіографії 1 Кбайт, максимальний – 8 Кбайт. Приведіть сутність до 1НФ

Співробітник

П.І.П/Б
Дата народження
Адреса
Телефон домашній
Телефон службовий
Телефон мобільний 1
Телефон мобільний 2

9. Приведіть сутність до 3НФ, якщо для кожного маршруту випуски однакові.

Маршрути

Номер
Випуск
Початкова_зуп
Кінцева_зуп
Довжина_пряма
Довжина_зворотна
Початок_роботи
Закінчення_роботи
Початок_перерви
Закінчення_перерви

10. Для яких даних можна використати поле об'єкта OLE? Вкажіть типи даних для кожного атрибута таблиці «Водій» бази даних.

Водій

Таб_номер
Прізвище
Ім'я
По батькові
Дата народження
Рід
Класність
Автобіографія
Фотокартка
Сімейний стан
Телефон

## **ПРАКТИЧНЕ ЗАНЯТТЯ №5**

### **Ознайомлення з CASE-технологіями. Побудова контекстної діаграми та функціональної моделі IDEF1**

З появою сучасної обчислювальної техніки виникли і досить інтенсивно розвиваються комп'ютерні засоби аналізу систем. Вони використовуються для побудови інформаційних систем і відомі як CASE-технології. Термін CASE (Computer Aided Software Engineering) використовується сьогодні досить широко. Під терміном CASE-засоби розуміють програмні засоби, які підтримують процеси аналізу та формулювання вимог до складних систем різного призначення, процеси створення і супроводу інформаційних систем, засоби проектування приладного програмного забезпечення.

Одним з напрямків цих технологій є методологія структурного аналізу та проектування – SADT-методологія (Structured Analysis and Design Technique). Цей напрямок розробки програмних засобів обчислювальної техніки виник в 60 – 70 роках минулого сторіччя і знайшов розвиток в роботах Дугласа Росса. Він широко використовується при вирішенні проблем аналізу проектування та експлуатації складних систем. На його основі розроблена методологія IDEF0 (Icam DEFinition), яка є головною частиною програми ICAM (Інтеграція комп'ютерних та промислових технологій), виконаної за ініціативою BBC (США). Методологію системного структурного аналізу використовує НАТО, МВФ (міжнародний валютний фонд), інші міжнародні організації та передові промислові фірми. У 90-х роках минулого сторіччя розроблені та прийняті міжнародні стандарти з використання SADT-технологій та побудови моделей складних систем відомих як IDEF-моделі. Ці моделі використовуються для розробки та вироблення стратегічних напрямків розвитку, для аналізу бізнес-процесів та ін. Характерними рисами вказаних технологій є :

- чітка мова опису систем, яка має універсальний характер;
- використання комп'ютерної техніки;

- стандартизація;
- регламентація усіх етапів виконання аналізу – від визначення проблеми до кінцевого продукту та його впровадження у практичну діяльність.

### 5.1. Опис функціональної моделі

Функціональна модель складається з контекстної діаграми та діаграм декомпозиції.

Контекстна діаграма зображується прямокутником з вхідними та вихідними величинами. В прямокутнику контекстної діаграми вказується основна функція (ціль системи). Входи та виходи контекстної діаграми розподілені не по двох, а по 4-х сторонах прямокутника. Призначення цих сторін такі:

- Ліва сторона відповідає входам системи (input), величинам, які надходять у систему і переробляються нею у для керування (control), тобто різним керівним діям, командам, стратегіям поведінки, процедурам, документам, що регламентують виконання роботи, тощо. Ці величини не змінюються, а служать тільки для керування.
- Права сторона відповідає виходам системи (output), продуктам її діяльності, результатам перетворення вхідних величин, шкідливим виділенням, відходам тощо.
- Нижня сторона відповідає механізмам (mechanism), а саме засобам, ресурсам, за допомогою яких виконуються вказані в прямокутнику функції.

Вигляд контекстної діаграми показано на рис. 9.

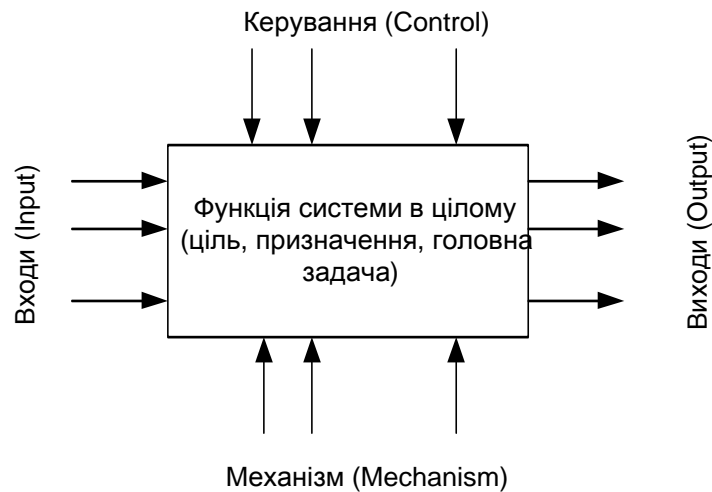


Рис. 9 – Загальний вигляд контекстної діаграми

Діаграми декомпозиції є подальшою деталізацією контекстної діаграми. Спочатку функція системи в цілому (ціль, призначення, головна задача) розбивається на декілька окремих функцій (завдань, робіт, цілей). Таких функцій рекомендується обирати від 2 до 6. Ці функції (їх деколи називають роботами – activity) зображують на окремому аркуші декомпозиції у вигляді функціональних блоків. Кожен функціональний блок (робота), в якості якого виступає окрема функція (робота, ціль чи завдання), зображують прямокутником. Сторони прямокутників робіт (функціональних блоків) мають таке ж призначення, що і розглянуті вище сторони контекстної діаграми. Між окремими функціональними блоками установлюють зв'язки, які відповідають логіці функціонування системи. Зв'язки між функціональними блоками зображують стрілками (Arrow) (дуже часто їх називають дугами). Кожна дуга (стрілка) відповідає передачі від блоку до блоку якогось конкретного об'єкта (предмета, речовини, документа, їх сукупності, а інколи і усного розпорядження) чи їх сукупності. Дуги можуть розгалужуватись та зливатись, як це показано на рис.10.



Рис. 10 – Розгалуження та злиття дуг

Дуга, яка утворилась при злитті інших дуг, відповідає усім об'єктам, які містяться у дугах до їх злиття. Після розгалуження дуга може відповідати усім об'єктам або частині з них. Для того, щоб вказати, яким саме об'єктам відповідає дуга, на ній ставлять мітку, присвоюють назву дузі та супроводжують описом об'єктів. Встановлення міток дуг, присвоєння назви дугам та опис об'єктів, яким вони відповідають, повинні задовольняти певним правилам, які будуть розглянути далі.

В моделі IDEF0 існує 5 типів зв'язків між функціональними блоками. Кожному з них відповідає певне розміщення дуг відносно блоків. Типи зв'язків такі:

- вхід – вихід;
- керування;
- вихід – механізм;
- зворотний зв'язок з керування,
- зворотний зв'язок з входу.

Вони зображені на рис. 11.

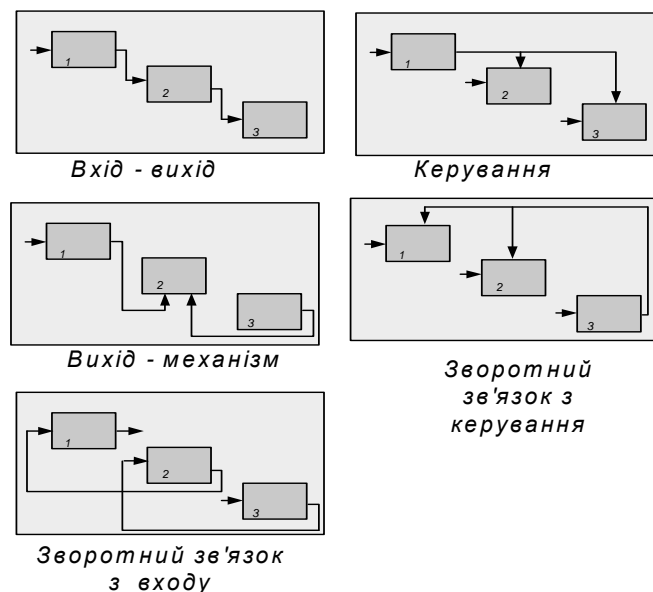


Рис. 11 – Типи зв'язків блоків функціональної моделі IDEF0



Фрагмент реальної моделі зі зворотним зв'язком з керування показано на рис. 12.

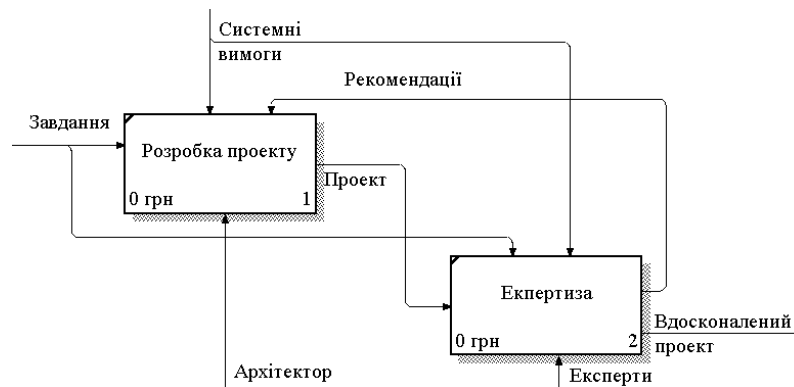


Рис. 12 – Фрагмент моделі «Розробка проекту»

## 5.2. Вимоги до порядку побудови моделі

1. Будь-яка SADT-модель вимагає точного визначення границь системи, цілей моделювання, точки зору, контексту розгляду системи.
2. SADT-моделі вимагають, щоб система розглядалась весь час з однієї точки зору, оскільки зміна точки зору може зробити модель неадекватною. Точка зору диктує підбір потрібної інформації та спосіб її подачі.
3. SADT-моделі будуються з верхнього рівня “з голови”. У них діаграми нижчого рівня є деталізованими діаграмами верхнього рівня. Кінцевий результат – це ієрархічна структура діаграм.

Під час побудови функціональної моделі необхідно чітко визначити:

- границі системи;
- ціль моделювання;
- контекст розгляду системи;
- точку зору;

тобто, відповісти на питання, які є вихідними будь-якого системного аналізу.

Мета є напрямком діяльності та критерієм закінченості моделі.

Визначаючи ціль, слід розрізняти:

- ціль системи, для якої будують функціональну модель;
- ціль моделювання.

Мета системи визначає зміст моделі її конкретне наповнення. Будь-яку модель ми будемо відповідно з її метою, із завданнями, які виконує система.

Метою моделювання може бути виконання певного завдання, вивчення системи, розробка та вдосконалення системи і т.п. Ця ціль визначає обсяг роботи, потрібний ступінь деталізації системи, розміри моделі в глибину і ширину.

Точка зору визначається тим, з погляду якої особи ми розглядаємо модель, точка зору якої особи найбільш повно відповідає меті моделювання.

Будь-яка модель повинна бути цілісна і розглядатись з однієї точки зору. Різні точки зору призводять до різних моделей, а зміна точки зору в ході моделювання не допускається, оскільки може призвести до невірному опису системи, до неадекватної моделі. Наприклад, система “тролейбусне депо”. Модель системи буде виглядати по-різному, якщо розглядати з різних точок зору. Для директора головними елементами є робочий колектив, підрозділи депо, виконання ними своїх функцій, одержання прибутку та оплата праці. Для головного інженера в основі знаходяться технологічні процеси роботи депо, їх технічне забезпечення, технічний рівень виконання операцій, використання наукових досягнень, технічний стан обладнання. Для економіста найбільш важливою є економічна ефективність роботи підрозділів, наявність фінансових ресурсів, одержання прибутку від діяльності. Для начальника служби охорони праці найбільш важливими є умови праці кожного робітника, обладнання робочих місць засобами захисту, організація профілактичної роботи з дотримання правил охорони праці і т.п. Отже, будь-яка система виглядає по-іншому з різних точок зору. Модель системи повинна виконуватись з однієї точки зору. Зміна точки зору в ході моделювання призводить до змішування і створення еклектичної моделі, яка не може дати відповіді на жодне питання. Для повного опису системи деколи виникає потреба розглядати її з різних точок зору, але у такому випадку необхідно будувати декілька моделей. В кожній моделі система розглядається тільки з однієї точки зору.

Під час побудови моделі при переході від одного рівня декомпозиції до іншого виникає потреба розглядати систему з точки зору виконавців конкретних функцій чи робіт. Така зміна точки зору потрібна і є корисною. Вона дозволяє одержати найбільш інформативну і корисну модель. Для забезпечення цілісності моделі тут потрібно на кожному етапі декомпозиції контролювати, як функції елементів відповідають загальній точці зору на модель, чи не змінюється ця точка зору. На кожному етапі моделювання загальна точка зору повинна залишатись незмінною, а змінюється тільки рівень деталізації.

Контекст розгляду системи визначає оточення, в якому ми розглядаємо систему. В попередніх розділах опису найпростіших типів моделей було розглянуто характеристики зовнішнього середовища. Контекст розгляду системи якраз і включає зовнішнє середовище з його характеристиками.

### **5.3. Побудова функціональної моделі системи (моделі IDEF0)**

На початкових етапах побудови функціональної моделі необхідно зібрати якомога більше інформації про систему, виконати усі попередні етапи системного аналізу, а саме: опис на вербальному рівні, історичний, морфологічний, функціональний аналіз та побудувати найпростіші моделі. Попередньо побудовані моделі “чорний ящик”, “склад системи”, “структура системи” та “структурна схема” значно полегшують побудову функціональної моделі.

Побудова функціональної моделі полягає в функціональній декомпозиції системи. Виконуючи декомпозицію, окремі функції системи розбивають на підфункції і зображують у вигляді ієрархічної структури. Під час аналізу систем, які існують, а не проектуються, найчастіше функціональну декомпозицію виконують на основі реальної структурної схеми системи. В першу чергу визначають функції окремих підсистем, і їм у відповідність ставлять функціональні блоки. Функціональні блоки необов'язково повинні відповідати структурним одиницям системи, оскільки кожна структурна

одиниця може виконувати декілька функцій або одну функцію може виконувати декілька підсистем. Допускається виконувати функціональну декомпозицію незалежно від структурної схеми системи, а орієнтуватись тільки на функції, які повинна виконувати кожна підсистема. Це часто роблять з метою реінжинірінгу систем, тобто для перебудови та вдосконалення роботи системи. Для систем, які проектуються, функціональна декомпозиція виконується раніше розробки структурної схеми, а структурна схема розробляється уже на основі функціональної моделі.

Як правило, орієнтовну декомпозицію системи бажано виконати під час підготовки до моделювання. Виділеним функціональним блокам системи слід присвоїти імена відповідно до їх функцій. Імена функціональних блоків повинні відображати функцію дієсловом, дієслівним зворотом або віддієслівним іменником. Присвоєння інших імен вважається помилкою. Результати функціональної декомпозиції, виконаної на початку побудови моделі, бажано зобразити на папері у вигляді дерева вузлів.

Наступним кроком є визначення вхідних та вихідних величин усіх функціональних блоків – робіт (згідно з прийнятою термінологією, англ. – activity). Тут необхідно розглянути величини, які надходять у кожен функціональний блок (роботу), які величини є результатами діяльності блоків, які забезпечують умови виконання функцій блоків і є механізмами та ресурсами для виконання функції кожним блоком. Слід мати на увазі, що, крім технологічних операцій, вказаних у найменуванні функціонального блоку, повинні відображатись і операції інформаційного характеру, керівні дії, тобто кожен блок повинен мати вхід з керування.

Після цього приступають до побудови функціональної моделі.

Функціональна модель являє собою низку діаграм, а саме контекстну діаграму, діаграми декомпозиції та деяких інших документів.

Елементи контекстної діаграми та діаграм декомпозиції такі:

- функціональні блоки (роботи) – activity;
- інтерфейсні дуги (стрілки) – arrow.

Функціональні блоки діаграм – роботи (activity) – зображують прямокутниками. Сторони прямокутника мають таке призначення:

- вхід (input);
- керування (control);
- вихід (output);
- механізм (mechanism).

Після побудови контекстної діаграми переходять до побудови наступних діаграм моделі. Наступні діаграми є діаграмами декомпозиції. Діаграма декомпозиції будується у такому порядку. Вибравши значок декомпозиції, у діалоговому вікні вказують тип діаграми, яка буде побудована, наступним кроком (пакет BPwin дозволяє будувати складні моделі з різними типами діаграм) потрібно вказати діаграму IDEF0 та кількість блоків декомпозиції. Діалогове вікно вибору кількості блоків декомпозиції та типу діаграми декомпозиції показане на рис. 13.



Рис. 13 – Діалогове вікно вибору кількості блоків декомпозиції та типу діаграми декомпозиції

Кількість блоків декомпозиції рекомендується вибирати в діапазоні від 2 до 6. Обмеження зумовлені тим, що не має сенсу декомпозиція, яка замінює один блок іншим, а число 6 рекомендоване тому, що діаграма, в якій більше 6 блоків та зв'язків між ними, стає досить складною і читати її важко. В

пакеті ВРwin допускається вводити до 8 блоків. Після вибору типу діаграм та кількості блоків робіт з'являється бланк діаграми декомпозиції на зразок, показаний на рис. 14.

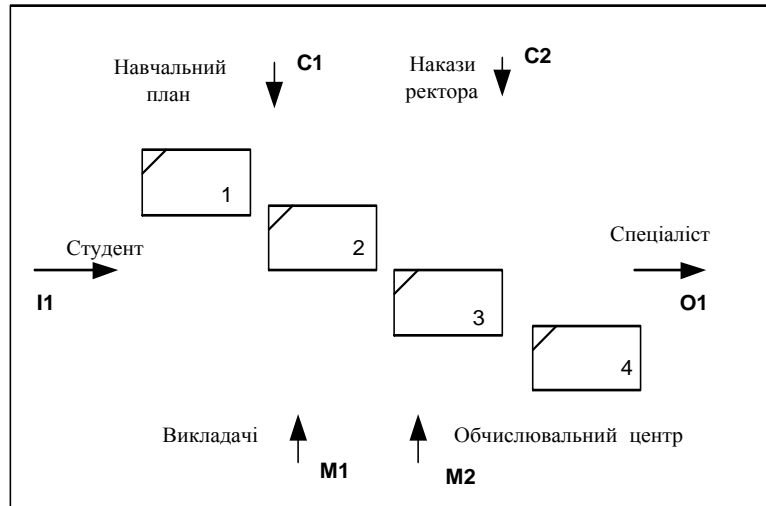


Рис. 14 – Зразок бланка декомпозиції

Бланк діаграми декомпозиції містить блоки робіт, розміщені зверху – вниз, зліва – направо. Використовуючи цей зразок, дамо пояснення змісту діаграм декомпозиції та правилам їх побудови. Розміщення функціональних блоків зверху – вниз, зліва – направо називають домінантним. У верхньому лівому кутку розміщується блок роботи, яку вважають найбільш пріоритетною, нижче і правіше знаходяться блоки менш пріоритетних функцій або у випадку, коли діаграма зображає поопераційне виконання певних робіт, то домінантність блоків визначає порядок виконання робіт, яким відповідають ці блоки. Блоки функціональної діаграми мають порядковий номер та закреслений лінією верхній кут. Нумерація блоків відповідає їх пріоритетності та слугує для встановлення єдиного порядку блоків у всій моделі. Закреслений лінією лівий верхній кут блоку є ознакою, що даний блок ще не підлягав операції декомпозиції. Якщо у подальшому виконати декомпозицію цього блоку, то позначка буде знята, якщо не виконувати, то вона залишиться на діаграмі.

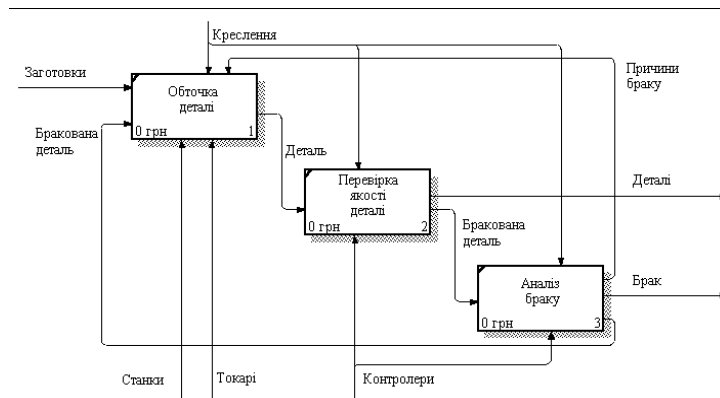


Рис. 15 – Діаграми декомпозиції системи “Токарна дільниця”

На рис. 14 і рис. 15 показано приклад побудови функціональної моделі системи “Токарна дільниця”. Модель складається із двох діаграм: контекстної і діаграми декомпозиції.

## **ПРАКТИЧНЕ ЗАНЯТТЯ №6**

### **Розробка запитів та звітів для аналізу даних ІС (АРМ)**

Ефективне використання баз даних зумовлене тим, що в них є прості та зручні у використанні засоби обробки інформації і одержання відповіді на будь-які запитання відносно розміщених даних. В СКБД таких засобів є декілька: сортування даних, використання фільтрів та використання запитів. Найбільш ефективним і універсальним є засіб запитів. За допомогою запитів можна відібрати дані, виконати потрібні розрахунки і представити результати вибору та розрахунків у вигляді таблиці значень. Результатом роботи програмного засобу вибірки є створення результуючої таблиці, у якій відображені потрібні за умовою запиту дані. Існує низка різних типів запитів: запит на відбір даних, запити на відновлення, зміну та вилучення даних, запити з розрахунками, запити з параметрами, перехресні запити та ін.

Перехресний запит відрізняється від звичайного тим, що у ньому результати аналізу групуються за двома ознаками та розміщуються у вигляді таблиці, на перехресті значень двох параметрів. Наприклад, помісячні пробіги кожної рухомої одиниці. Ознаками, за якими повинні групуватись дані, є номер рухомої одиниці та місяць. В результаті виконання такого запиту буде одержана таблиця, в якій рядки відповідають номерам РО, а – колонки місяцям, за які розраховано пробіги. Результати розрахунків, тобто пробіги РО, будуть знаходитись в клітинках таблиці, на перехресті значень номера РО та місяця.

Перехресний запит створюється на основі однієї таблиці чи запиту. Для створення перехресного запиту на основі декількох таблиць спочатку потрібно створити звичайний запит, в якому будуть усі потрібні поля.

Запити є основними формами роботи з базами даних. Вони дозволяють вирішити завдання, які постають перед ІС (чи АРМом).

Основне призначення звітів – це видача інформації в роздрукованому вигляді. Таблиці, запити і форми бази даних також можна роздрукувати. На



відміну від запитів і форм, звіти дозволяють професійно оформити документи з використанням різних типів шрифтів, виконувати розподіл документів на розділи, використовувати заголовки, зауваження, рисунки, діаграми, колонтитули і т.п. Звіти дозволяють безпосередньо друкувати результати аналізу на принтері або зберігати їх у вигляді текстового файлу.

Звіти забезпечують виконання складних обчислень за даними таблиць і групування даних за різними ознаками до 10 рівнів ієрархічної структури.

Запити та звіти створюються відповідно до методики, що засвоюється під час виконання лабораторних робіт. Порядок розробки наведений у відповідних методичних вказівках.

### **Контрольні питання**

1. Як вивести у запиті добовий пробіг тролейбусів?
2. Як вивести у запиті місячний пробіг тролейбусів?
3. Як вивести у запиті пробіг тролейбусів по різних маршрутах?
4. Як вивести у запиті загальний пробіг тролейбусів депо за місяць?
5. Як вивести у запиті роботу маршруту за місяць у машино-кілометрах?
6. База даних містить 100 таблиць, середнім розміром 500 Кбайт. Розмір одного запису в середньому 1000 Кілобіт. Розрахуйте завантаженість мережі, якщо у ній одночасно користується базою даних 100 комп'ютерів, які роблять в середньому по 3 запити за хвилину. Мережа працює по архітектурі файл-сервер. Службова інформація складає 20%, що передається за допомогою запитів.
7. Розрахуйте завантаженість цієї мережі, якщо вона працює по архітектурі клієнт-сервер і службова інформація складає 120%, що передається запитам.

## ПРАКТИЧНЕ ЗАНЯТТЯ №7

### Розробка форм, автоматизація створеної моделі ІС (АРМ)

Розглянуті таблиці та запити є ефективними засобами розробки ІС чи АРМ на основі бази даних. СКБД Access представляє широкі можливості роботи з таблицями і запитами. З частиною цих можливостей ви познайомились під час виконання лабораторних робіт. Використання таблиць та запитів зручне для того, хто розробляє певну ІС чи АРМ, працює з СКБД в діалоговому режимі.

Користувача ІС чи АРМа питання структури таблиць, зв'язків між ними, нормалізації даних і т.п. не цікавлять. Йому потрібні прості в користуванні та зрозумілі засоби роботи з даними. Тобто потрібен зручний у користуванні й простий інтерфейс. Для розробки такого інтерфейсу призначені форми та звіти.

Форми забезпечують зручні засоби вводу, заміни, збереження даних, відбору їх та видачі користувачу результатів виконаних розрахунків.

Форми можуть створюватись для виконання таких завдань:

- Перегляду і зміни даних.
- Вводу нових даних.
- Друкування результатів.
- Видачі повідомлень користувачу під час роботи.

Сама форма може мати вигляд зошита зі сторінками (***В столбце***), рулону з записами (***Ленточная***), у вигляді таблиці (***Табличная***), а також у вигляді діаграми.

Під час перегляду даних пов'язаних таблиць досить часто необхідно, щоб у формі відображались дані декількох таблиць. Так, наприклад, при проектуванні роботи тролейбусів на маршрутах потрібно розглянути усі випуски кожного конкретного маршруту. Маршрути і випуски розміщені в окремих таблицях, між якими встановлено зв'язок типу 1:∞. Кожному маршруту відповідає низку випусків. Для перегляду випусків у цьому випадку зручно створити форму, яка в собі містить підпорядковану. Підпорядкована форма використовується,

наприклад, якщо потрібно розглянути дані з пов'язаних таблиць у разі зв'язку «один до багатьох».

Форми створюються на основі таблиці чи запиту (пов'язаних таблиць).

Форма керування роботою створюваної ІС чи АРМ розробляється по завершенні створення усіх потрібних запитів, форм та звітів. Форми усіх типів створюються відповідно до методики, що засвоюється під час виконання лабораторних робіт. Порядок розробки наведений у відповідних методичних вказівках.

## ПРАКТИЧНЕ ЗАНЯТТЯ №8

### Створення презентацій для демонстрації роботи створеної ІС (АРМ)

Порядок створення презентацій наведено нижче.

Відкрити за допомогою файлового менеджера “Провідник” чи іншого, наявного на комп’ютері, власний каталог, у якому знаходиться розроблена ІС (чи АРМ), та розмістити у цьому каталозі усі матеріали, потрібні для побудови презентації. Створити необхідні файли з окремими діаграмами, зображеннями моделі, звітами по ній.

Завантажити пакет *POWER POINT* 

У початковому вікні створення презентації натиснути кнопку **Шаблон оформлення**, як показано на малюнку, а потім – ОК.

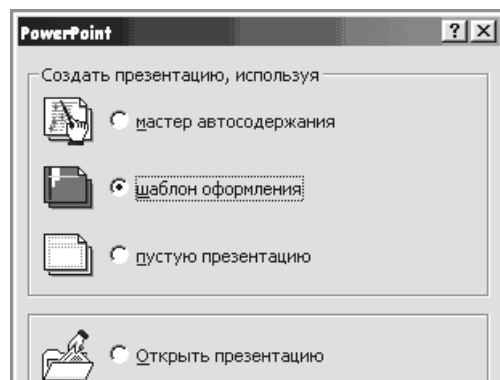


Рис. 16 – Меню Power Point

При відсутності вікна скористатись пунктом меню **Файл > Создать...**

У вікні **Создать презентацию** вибрати заставку **Шаблоны оформления** та бажаний варіант оформлення презентації. Підтвердити вибір натисканням клавіші **ОК**.

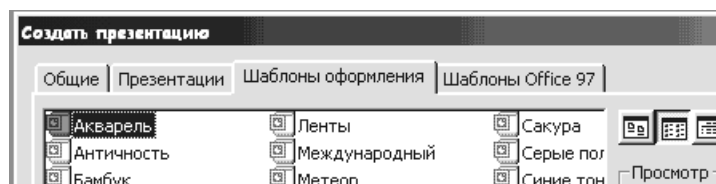


Рис. 17 – Вибір шаблону оформлення

У вікні *Создание слайда* Вибрати розмітку першого слайда – слайда титульної сторінки. Підтвердіть вибір клавішею **ОК**.



Рис. 18 – Меню створення слайда

Збережіть створений початок презентації у власному каталозі, скориставшись інструментом *Сохранить* чи пунктом меню *Файл > Сохранить...*

### 8.1. Створення першої сторінки презентації

У вікні авторозмітки титульної сторінки поставити курсор на поле заголовка та виділити його натисканням лівої клавіші мишки.

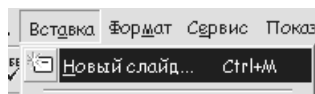
Введіть текст заголовка з назвою системи, яка є вашим варіантом розрахунково-графічної роботи.

Поставити курсор на підзаголовок та аналогічним чином введіть текст: Розрахунково-графічна робота з дисципліни “Інформаційні системи на транспорті”, виконана студентом 5 курсу факультету ЕТ, групи ... «Прізвище та ініціали».

Переглянути слайд та, у випадку необхідності, відредагувати його, ввівши вирівнювання по центру, потрібний шрифт та його розмір.

## 8.2. Створення наступних слайдів презентації

Вставити новий слайд, використовуючи панель інструментів **Новий слайд** чи пункт меню: **Вставка > Новый слайд**.



Вибрати потрібну авторозмітку слайда залежно від того, чи бажаєте вставити тільки текст чи ілюстрацію з текстом.

Введіть в слайд потрібний текст чи скопіюйте його з підготовленого раніше файла.

Установіть курсор на місці розміщення ілюстрації.

Вставте у слайд потрібну ілюстрацію, вибравши пункт меню **Вставка > Рисунок > Из файла...** та потрібний файл у вікні пошуку файла.

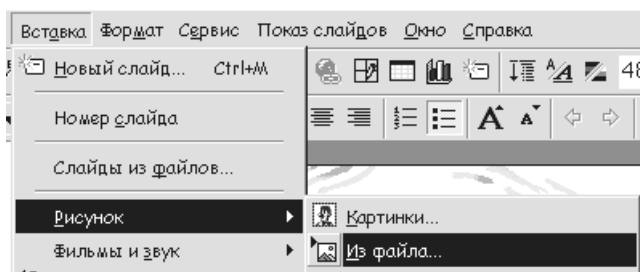


Рис. 19 – Меню вставки рисунка


Відредагуйте слайд, встановивши потрібний розмір ілюстрації, розмістивши її на слайді у потрібному місті та вибравши потрібний розмір шрифтів і розміщення тексту. Розмір ілюстрації можна змінити, перемістивши мишкою її край до бажаної позиції. Аналогічно можна змінити розмір тексту і потім відредагувати, вибравши бажаний шрифт.

## 8.3. Створення сторінки без авторозмітки

Вставте новий слайд. Під час вибору авторозмітки вкажіть у вікні **Создание слайда** пустий слайд, який не має авторозмітки.

Введіть художньо оформлений заголовок слайда, скориставшись панеллю інструментів **WORD ART**. Якщо така панель відсутня, то виведіть її за допомогою команди **Вид > Панели инструментов > WORD ART**.



Для введення заголовка слайда виділіть початкову кнопку панелі **WORD ART**  **Добавить объект WORD ART.**

Виберіть бажане оформлення тексту та введіть потрібний текст у вікно введення тексту з колекції **WORD ART**.




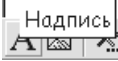
Введіть потрібний текст заголовка.

Введений текст, який з'явиться як заголовок слайда, розмістіть в потрібному місці та оберіть необхідний розмір.

Відредагуйте забарвлення зображення та вирівнювання тексту, для цього скористуйтеся інструментами з панелі **WORD ART** за зразком, показаним, на рисунку.



Введіть прямокутник (чи еліпс) для розміщення у ньому тексту. Для цього скористуйтеся панеллю інструментів **Рисование**. При відсутності такої панелі на екрані виведіть її за допомогою команди меню **Вид > Панели инструментов > Рисование**. Вибрати у цій панелі інструмент **Прямоугольник**  та вкажіть на поверхні слайда бажане місце його розміщення.

Виділіть прямокутник курсором, оберіть інструмент **Надпись**  та введіть потрібний текст у створений прямокутник.

**Зауваження 1:** Тексти на слайд слід вводити у найбільш лаконічній та чіткій формі, наприклад, розбивши на пункти та вводячи окремі пункти тексту.

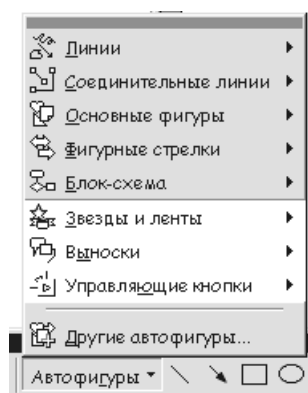
Відредагуйте розміри прямокутника з текстом так, щоб текст повністю розміщувався усередині прямокутника, і прямокутник мав бажане

співвідношення сторін, оберіть потрібне забарвлення фону прямокутника та надпису тексту.

Вставте у слайд потрібні зображення та графіки, наприклад, меню бази даних, таблиці, зображення логічної структури тощо. Одержати зображення можна, скориставшись клавішею **Print Screen** клавіатури, чи створивши зображення, наприклад, в графічному редакторі VISIO. Вставку після натискання клавіші **Print Screen** слід робити за допомогою контекстного меню правої клавіші мишки, пункт **Вставка**.

**Зауваження 2.** Пакет MS OFFICE допускає різні способи вставки, а саме: вставка з можливістю редагування малюнка у тому редакторі, в якому він був створений, вставка із взаємозв'язком з оригіналом, при якій на вставленому об'єкті відображаються усі зміни, котрі виконуються на оригіналі, а також вставка незалежного рисунка, відірваного від оригіналу. Перші з двох вказаних способів вставки суттєво збільшують розмір документа презентації і обсяг пам'яті, потрібної для її розміщення. Тому для зменшення розміру презентації бажано використовувати третій спосіб вставки, а саме вставку через меню **Правка > Специальная вставка**.

Вставте у слайд елементи оформлення у вигляд автофігур, скориставшись пунктом меню **Автофигуры**.

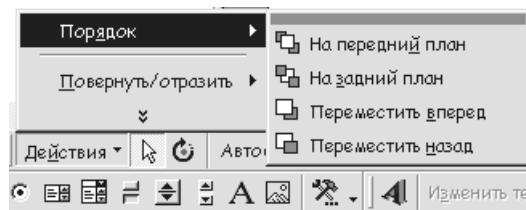




Для подальшого оформлення слайда вставте малюнки з бібліотеки пакета MS OFFICE або власних малюнків. Для використання бібліотеки скористайтесь пунктом меню **Вставка > Рисунок > Картинки**.

Відредагуйте зображення слайда, змінюючи розміри об'єктів та переміщуючи їх.

Встановіть порядок розміщення об'єктів, перемістивши їх на передній чи задній план. Для цього виділіть об'єкт та скористайтесь інструментом **Действия > Порядок**.



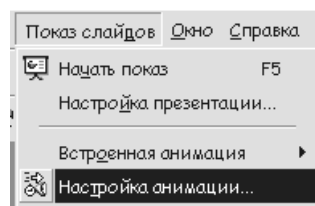
Закінчіть створення слайда і перейдіть до побудови наступного.

Закінчивши побудову всіх слайдів, збережіть створену презентацію.

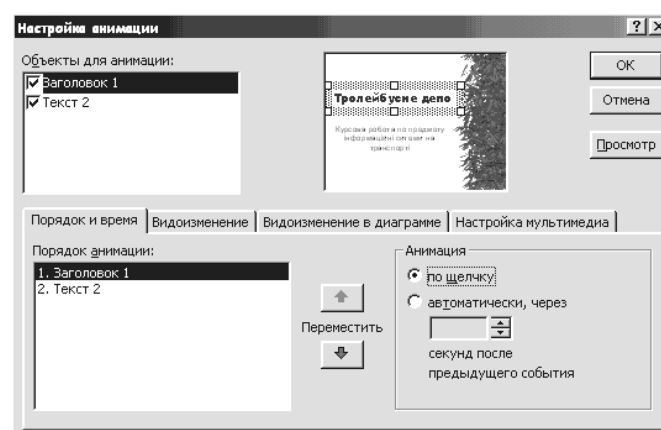
#### 8.4. Підготовка презентації до показу. Анімація слайда

Встановіть на екрані за допомогою клавіатури (клавіші **Page Up**) початковий слайд презентації.

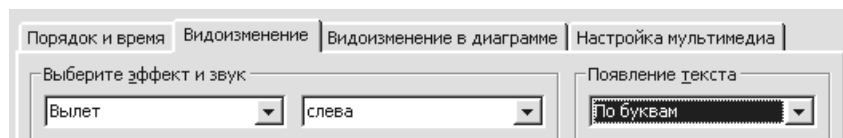
Вибрати пункт меню **Показ слайдов > Настройка анимации**.



У вікні **Настройка анимации** оберіть ярличок **Порядок и время** та виділити як об'єкт для анімації **Заголовок 1** (в пакеті MS OFFICE ярличок **Время**).



Вибрати ярличок **Видоизменения** та оберіть бажаний ефект зміни даного об'єкта, наприклад, **Вылет > Слева > По буквам**.



Вибрати наступний об'єкт **Текст 2** та вказати бажаний ефект зміни.

Натиснути кнопку **Просмотр** (див. попередній малюнок) та проглянути у прискореному темпі порядок показу даного слайда. У разі потреби повернутись та покращити ефект анімації.

Перейти до наступного слайда клавішею **PAGE DOWN** та вказаним чином введіть анімацію потрібних об'єктів.

Проглянути щойно створену анімацію. Змінити її, досягаючи бажаного ефекту.

## Контрольні питання

1. У таблиці бази даних 500 тис. записів. Як зміниться розмір таблиці, якщо для кожного з 8 символічних полів ввести обмеження допустимого розміру до 50 символів.
2. Розбийте зв'язок багато до багатьох щодо закріплення водіїв у депо за тролейбусами на два допустимі зв'язки у базі даних ACCESS.

Водій
Таб_номер
Прізвище
Ім'я
По батькові

Тролейбус
Рег_номер
Тип
Дата виготовлення
Дата вводу в експлуатацію

3. Запишіть формулу, як розрахувати довжину пробігу тролейбуса, якщо в базі даних є таблиця нарядів з вказівкою маршрутів і кількістю рейсів та таблиця маршрутів з довжиною зворотного рейсу та нульового пробігу

## ПРАКТИЧНЕ ЗАНЯТТЯ № 9

### Реалізація АРМ в комп'ютерній мережі

#### 9.1. Архітектура комп'ютерних мереж

Існує дві архітектури комп'ютерних мереж:

файл-сервер

клієнт-сервер

Архітектура файл-сервер – це така організація роботи ІС, в якій клієнтам на їх запити передаються файли інформації без обробки сервером.

Історично існувала першою. Вона найбільш проста. Програмне забезпечення не передбачає обробки інформації, а тільки пошук та передачу файлів.

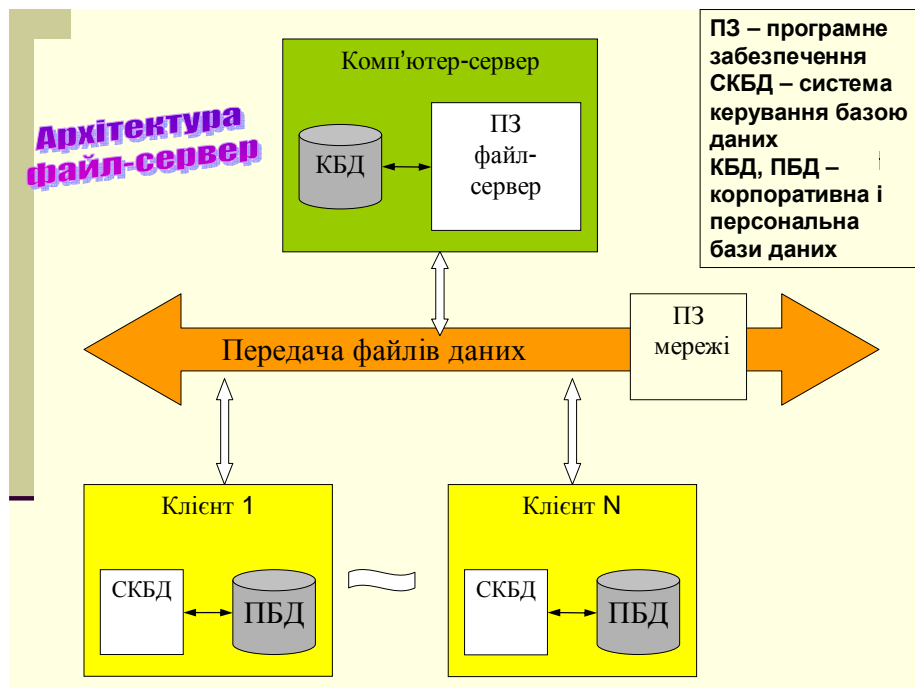


Рис. 20 – Архітектура комп'ютерної мережі файл-сервер

Переваги:

відносна простота програмного забезпечення;

захищеність інформації від несанкціонованого втручання.

Недоліки:

висока інтенсивність передачі даних;

завантаження ліній зв'язку.

Лініями зв'язку передається інформація, більшість якої не потрібна клієнтові.

Архітектура клієнт-сервер.

В лінії зв'язку передається тільки відібрана інформація. Комп'ютер-сервер обробляє запити клієнтів та видає тільки оброблену інформацію.

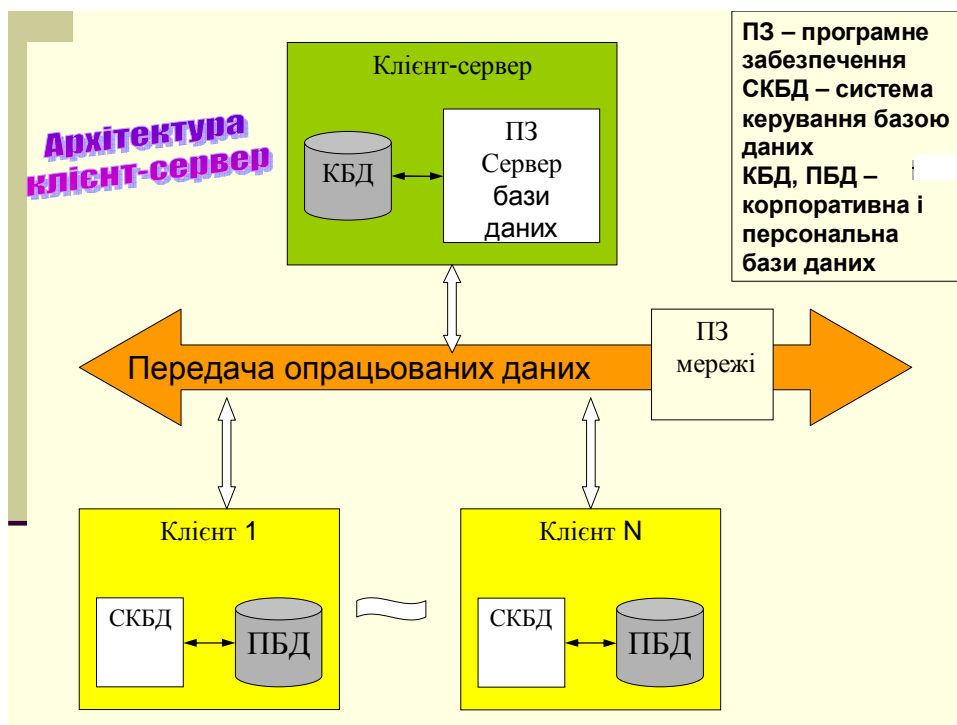


Рис. 21 – Архітектура комп'ютерної мережі клієнт-сервер

Переваги такої архітектури ІС:

- розподіл ресурсів між комп'ютерами;
- зменшення завантаження ліній зв'язку;
- вдале використання корпоративної та персональних баз даних.

Недоліки:

- ускладнення програмного забезпечення;
- небезпека несанкціонованого використання даних, їх пошкодження;
- ускладнення методів захисту та доступу до інформації.

## 9.2. Стандарти рівнів відкритих телекомунікаційних систем

Відкрита телекомунікаційна система (ВТС) – це система, яка надає різноманітні послуги, перетворюючи повідомлення в сигнал і навпаки. Вона – проміжний елемент між прикладним процесом і фізичним середовищем. В

своїй роботі з аналогічними собі використовує один і той самий набір стандартів.

Для зовнішнього користувача ВТС представлена одним стандартним чином.

Вимоги до цифрової мережі зв'язку (транспортні зобов'язання мережі):

Доставка до пункту призначення.

Мінімальний час доставки.

Мінімальні відмінності копії від оригіналу.

Рівні ВТМ визначаються стандартними процедурами, які потрібно виконати для забезпечення найрізноманітніших прикладних процесів.

Основні групи процедур такі:

Інформаційні

Транспортні

Мережеві

Для виконання цих процедур розробляють спеціальні засоби і програмне забезпечення, протоколи та інтерфейси. Користувачу не потрібно розуміти суть кожної процедури, але розробник ІС повинен розуміти принаймні суть процесів.

Стандартом OSI 7498, прийнятим в 1983 році, встановлено такі рівні відкритих телекомунікаційних систем:

1. Фізичний
2. Канальний рівень (рівень передачі даних)
3. Мережевий рівень
4. Транспортний рівень
5. Сеансовий рівень
6. Представлення даних
7. Прикладний рівень

Архітектура ВТС показана на рис. 22

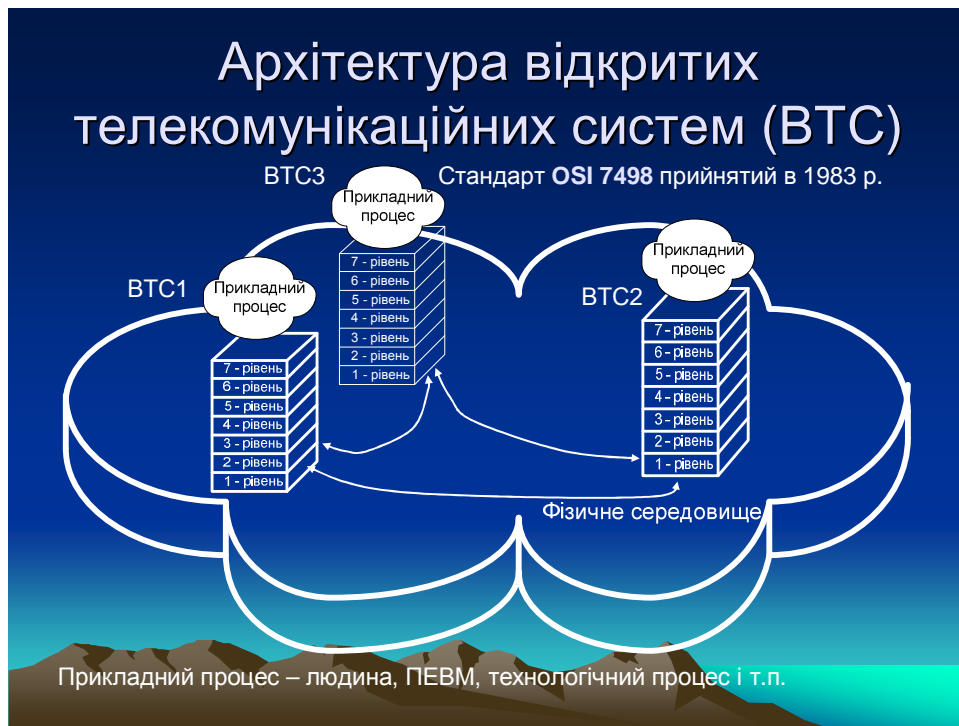
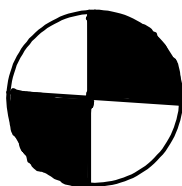


Рис. 22 – Архітектура відкритих телекомунікаційних систем (Інтернет)

## Контрольні питання

1. У фототелеграфі (факсі) світловий промінь пробігає один рядок аркуша паперу за 1 с. Розмір світлової плями 1 мм. Розрахуйте, за який час буде передано аркуш документа стандартного розміру (210 x 297 мм) і якій швидкості передачі даних в біт/с відповідає робота фототелеграфного апарата, якщо він розрізняє 16 відтінків зображення.
2. Для регулювання частоти обертання вала телеграфного апарата використовують камертон з частотою 412 Гц. Через щілину, яка відкривається при коливанні камертона, розглядають хвостовик вала з зображенням у вигляді круга поділеного на 4 частини. З якою швидкістю обертається вал апарата, якщо зображення здається нерухомим.



3. Стартстопний телеграфний апарат передає за допомогою коду Бодо до 200 символів за хвилину. Розрахуйте швидкість передачі інформації в лінії зв'язку в бітах за секунду.
4. Глядач знаходиться в концертному залі на відстані 100 м від артиста. Інший глядач дивиться передачу по телевізору. Хто раніше почує голос артиста, якщо телевізійний сигнал передається через геостаціонарний супутник, розміщений на висоті 35 тис. км. Визначити час затримки сигналу в обох випадках. Затримкою в ретрансляції сигналу нехтувати.
5. Прийнято послідовність 01111110110011111011001001111110. Якій відправленій послідовності вона відповідає? Чи вірна ця послідовність, якщо останній біт повідомлення є бітом парності?
6. Від клавіатури комп'ютера прийнято такі послідовності:
  1. 10010011
  2. 10100100
  3. 11110001
  4. 01010101
  5. 10101010
  6. 10010010
  7. 11111001
  8. 00010001

Які з них комп'ютер прийме, які забракує?

7. IP-адреса має наступну структуру:

xxxxxxxx.xxxxxxxxxx.xxxxxxxxxx.

Тут x – двійковий знак 1 чи 0. Скільки різних адрес може бути створено.

8. Прийнято таку послідовність символів:

0111111010010011100100111010111110

Визначити, чи вірно передані дані, якщо контрольною сумою є 101. В послідовність включені «прапори» – границі кадру.

9. Розрахувати довжини хвиль в частотних діапазонах супутникового зв'язку.

# Навчальне видання

Методичні вказівки  
до проведення практичних занять  
та самостійної роботи студентів  
з дисципліни

## «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ НА ТРАНСПОРТІ»

(для студентів усіх форм навчання спеціальностей:  
7.05070202, 8.05070202 «Електричні системи і комплекси  
транспортних засобів»; 7.05070203, 8.05070203 «Електричний транспорт»  
та слухачів другої вищої освіти спеціальності «Електричний транспорт»)

Укладачі: **СОРОКА** Костянтин Олексійович,  
**ГАРБУЗ** Нонна Володимирівна,  
**ГОРДІЄНКО** Ольга Сергіївна

Відповідальний за випуск *К. О. Сорока*

Редактор *О. В. Тарасюк*

Комп'ютерне верстання *І. В. Волосожарова*

План 2010, поз. 198М

---

Підп. до друку 17.12.2010 р.	Формат 60×84/16
Друк на ризографі.	Ум. друк. арк. 2,5
Зам. №	Тираж 100 пр.

Видавець і виготовлювач:  
Харківська національна академія міського господарства,  
вул. Революції, 12, Харків, 61002  
Електронна адреса: [rectorat@ksame.kharkov.ua](mailto:rectorat@ksame.kharkov.ua)  
Свідоцтво суб'єкта видавничої справи:  
ДК № 4064 від 12.05.2011 р.